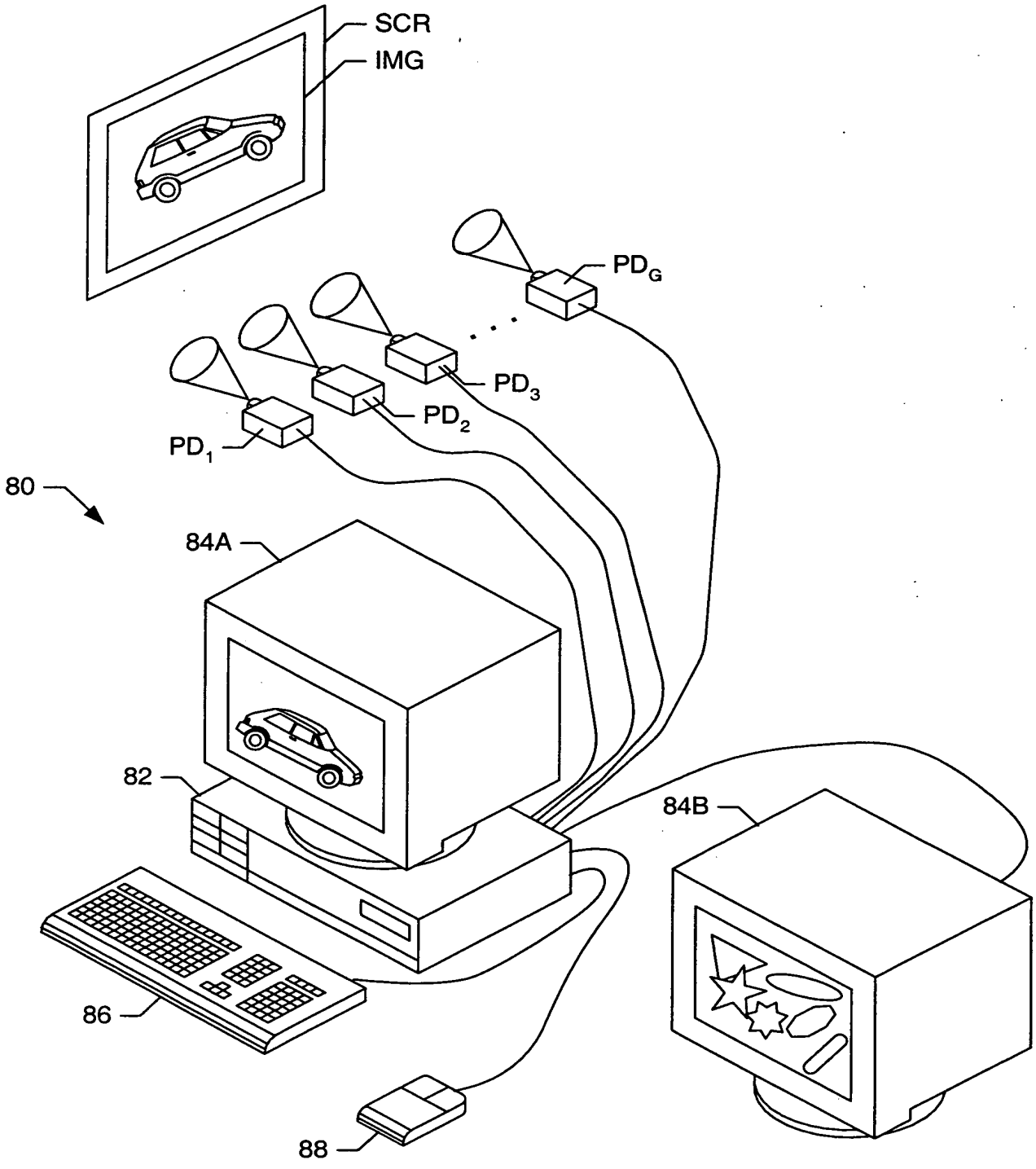


Fig. 1

DO6227 "262256D

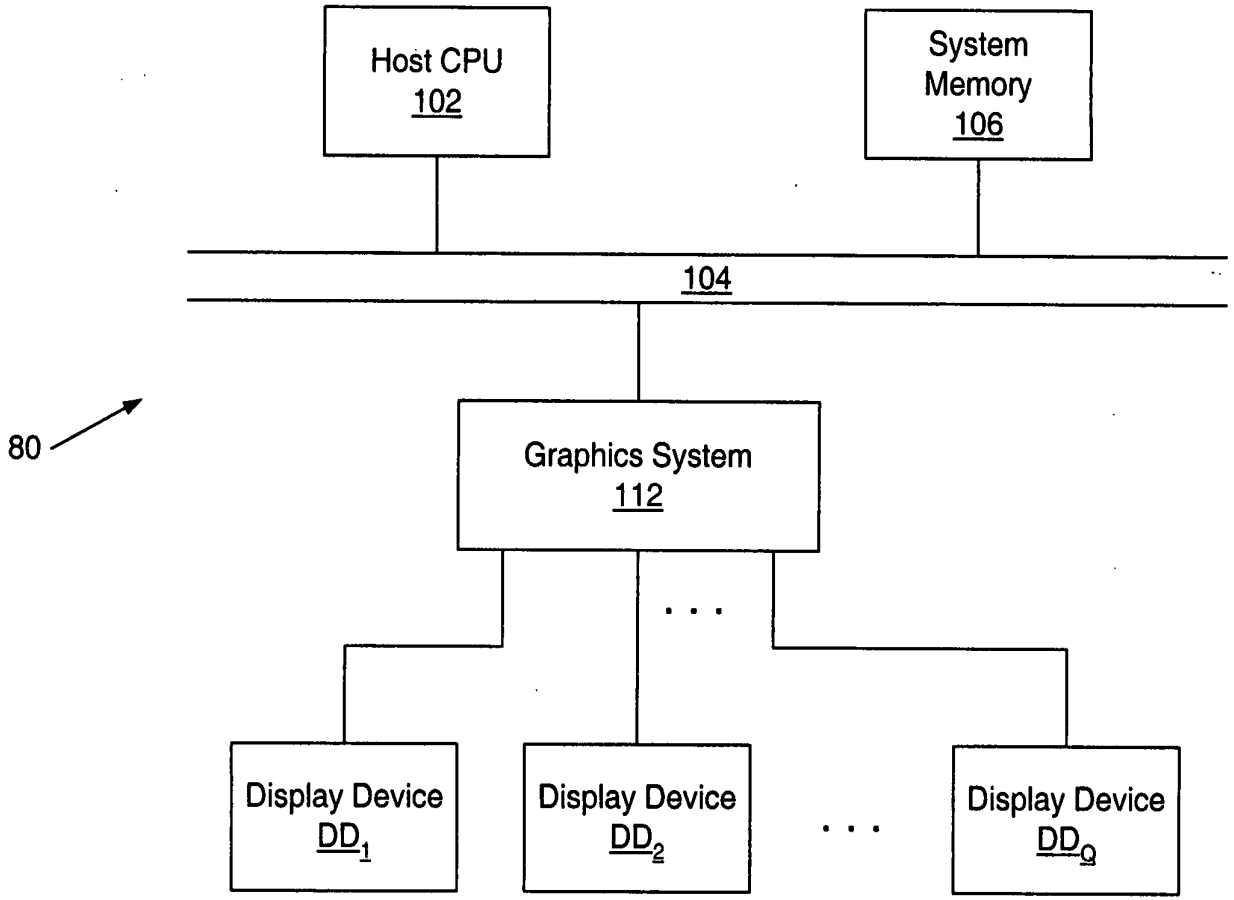


FIG. 2

00621T" /62256/60

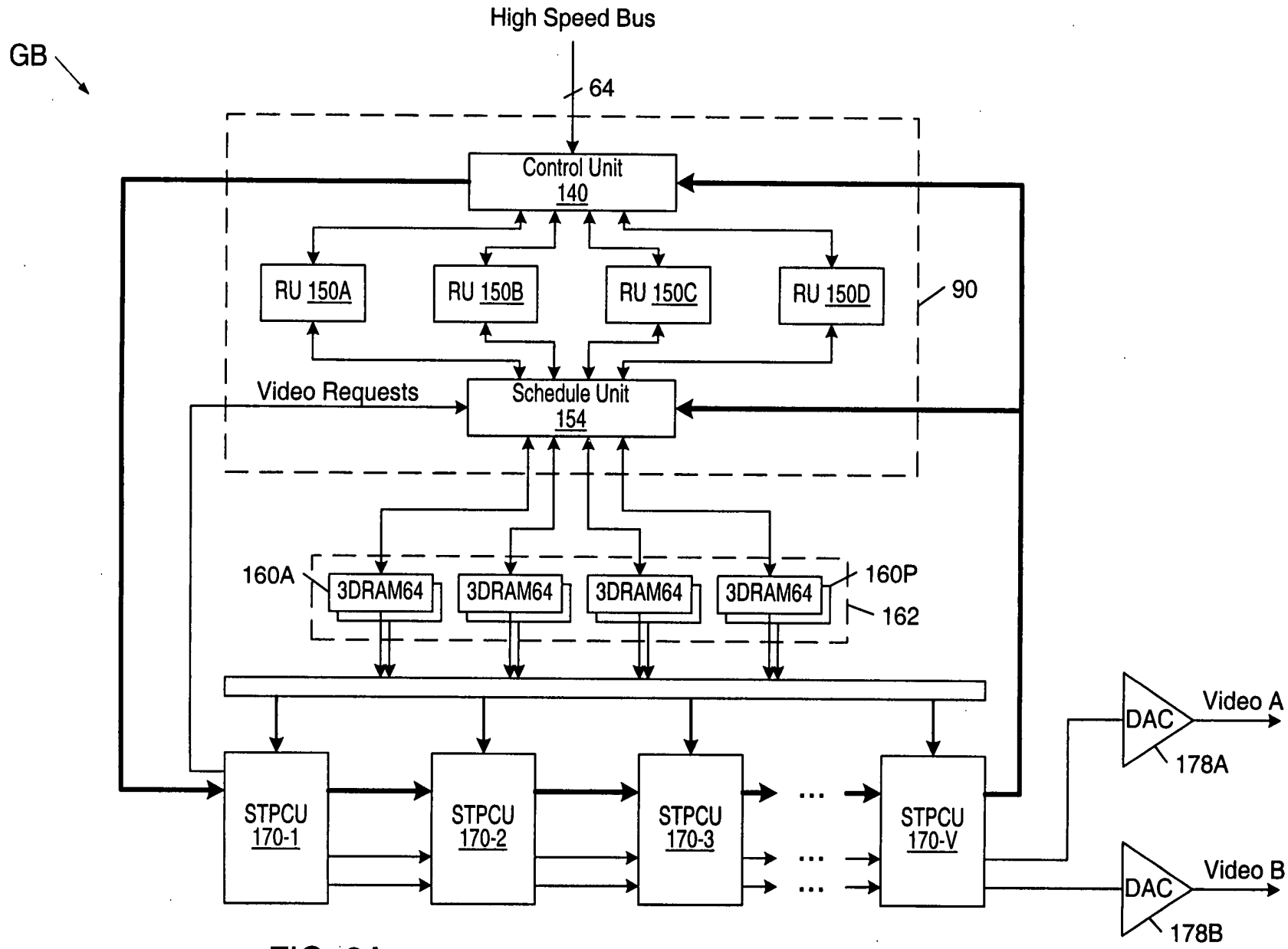


FIG. 3A

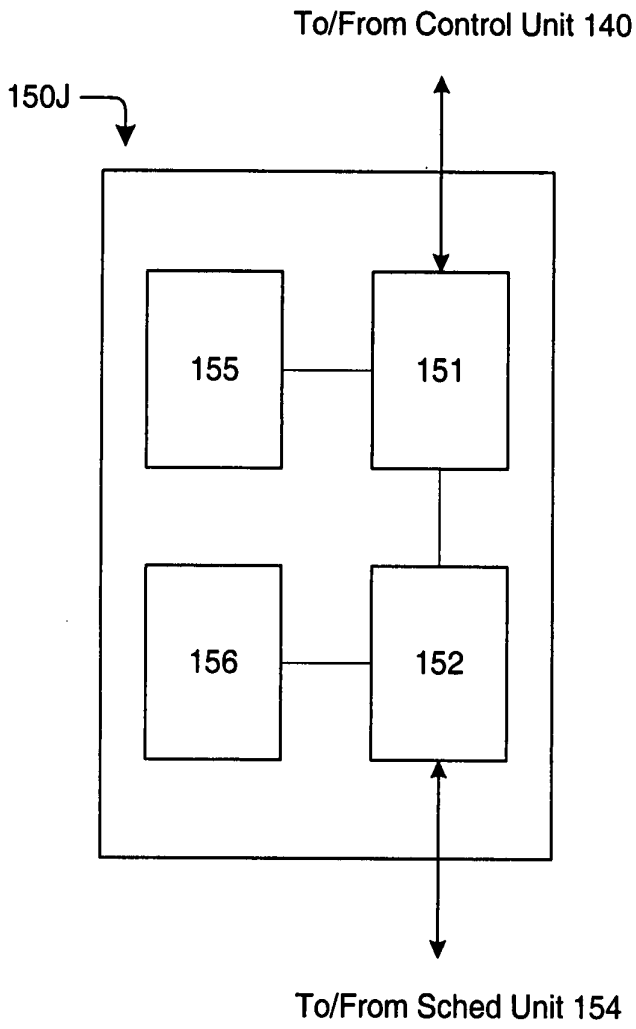

To/From Control Unit 140
150J
155
151
156
152
To/From Sched Unit 154

FIG. 3B

09627" 26325260

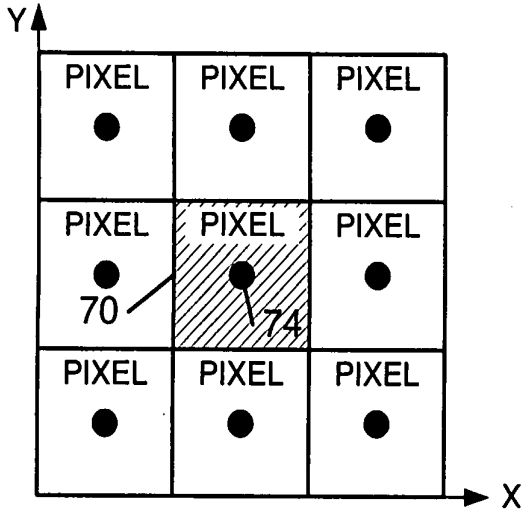


FIG. 4

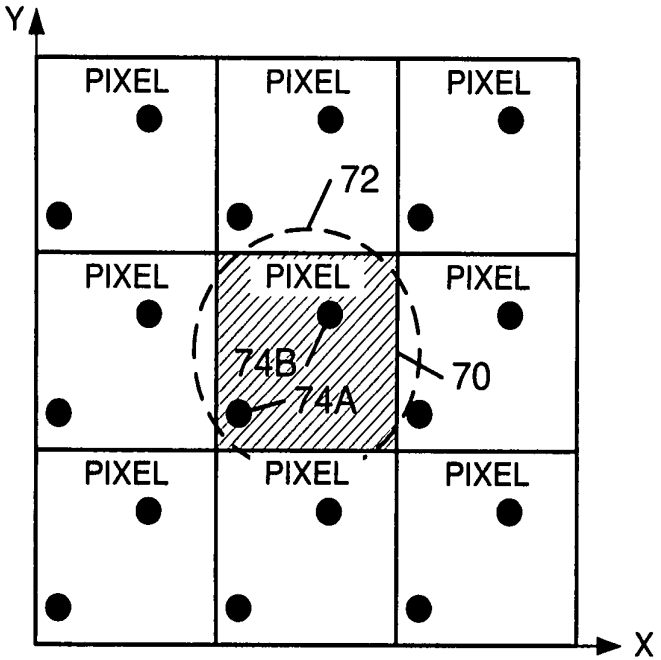


FIG. 5A

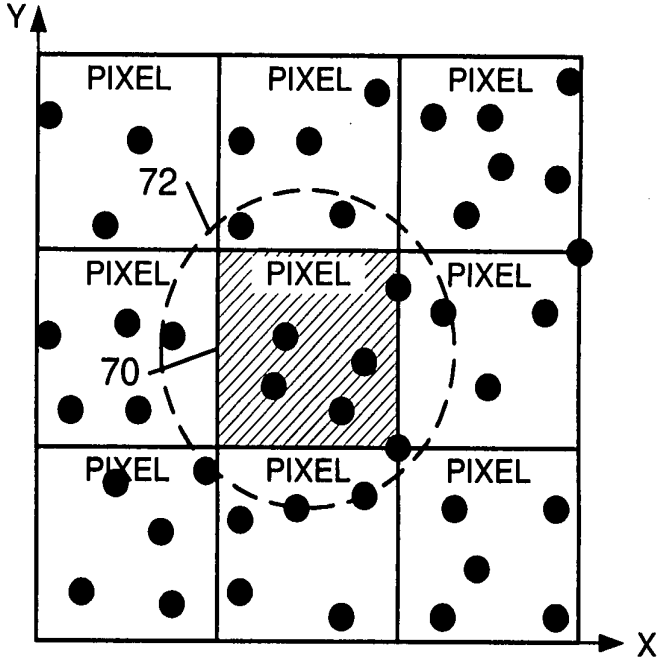


FIG. 5B

006227 "26325760

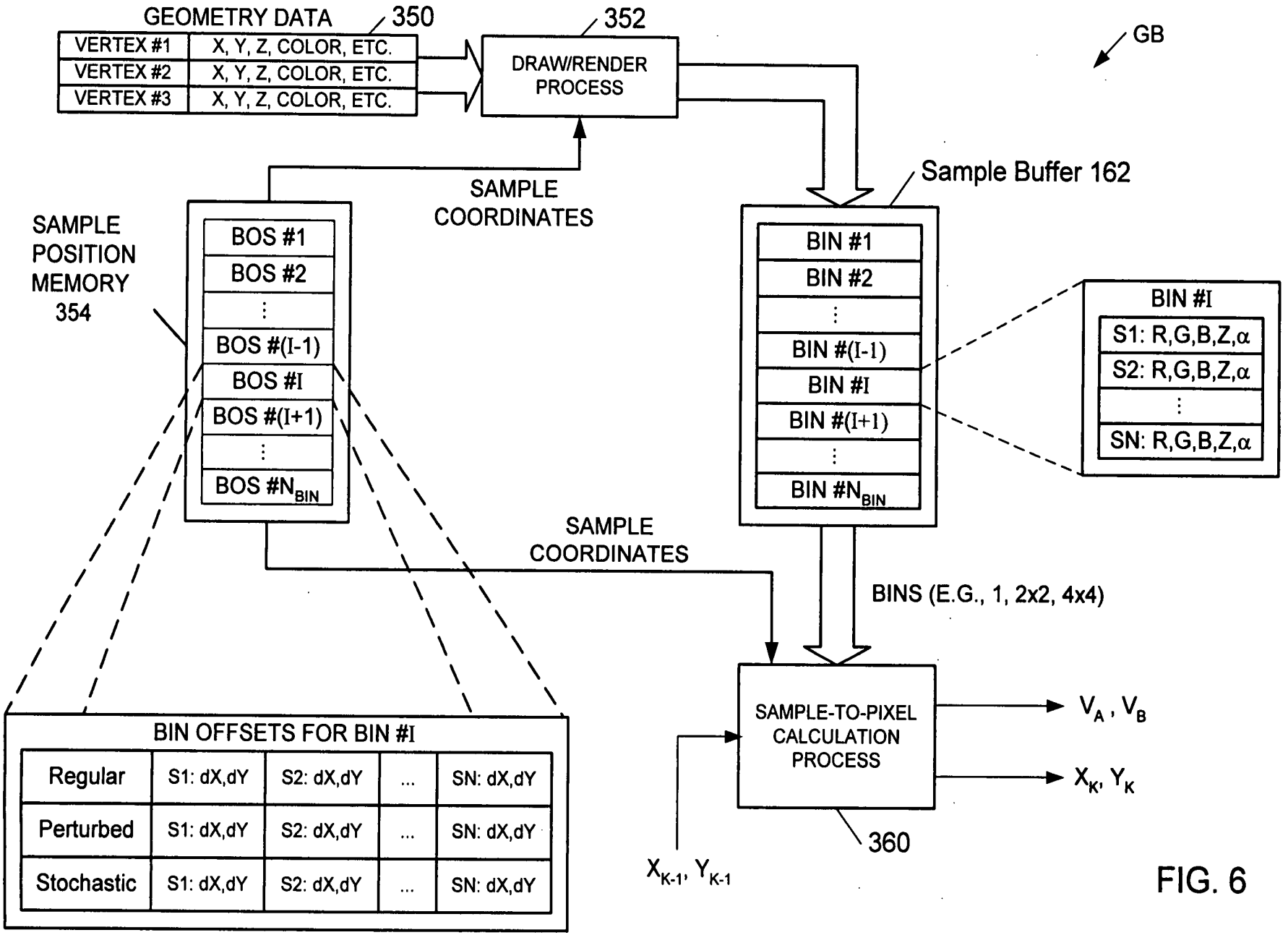


FIG. 6

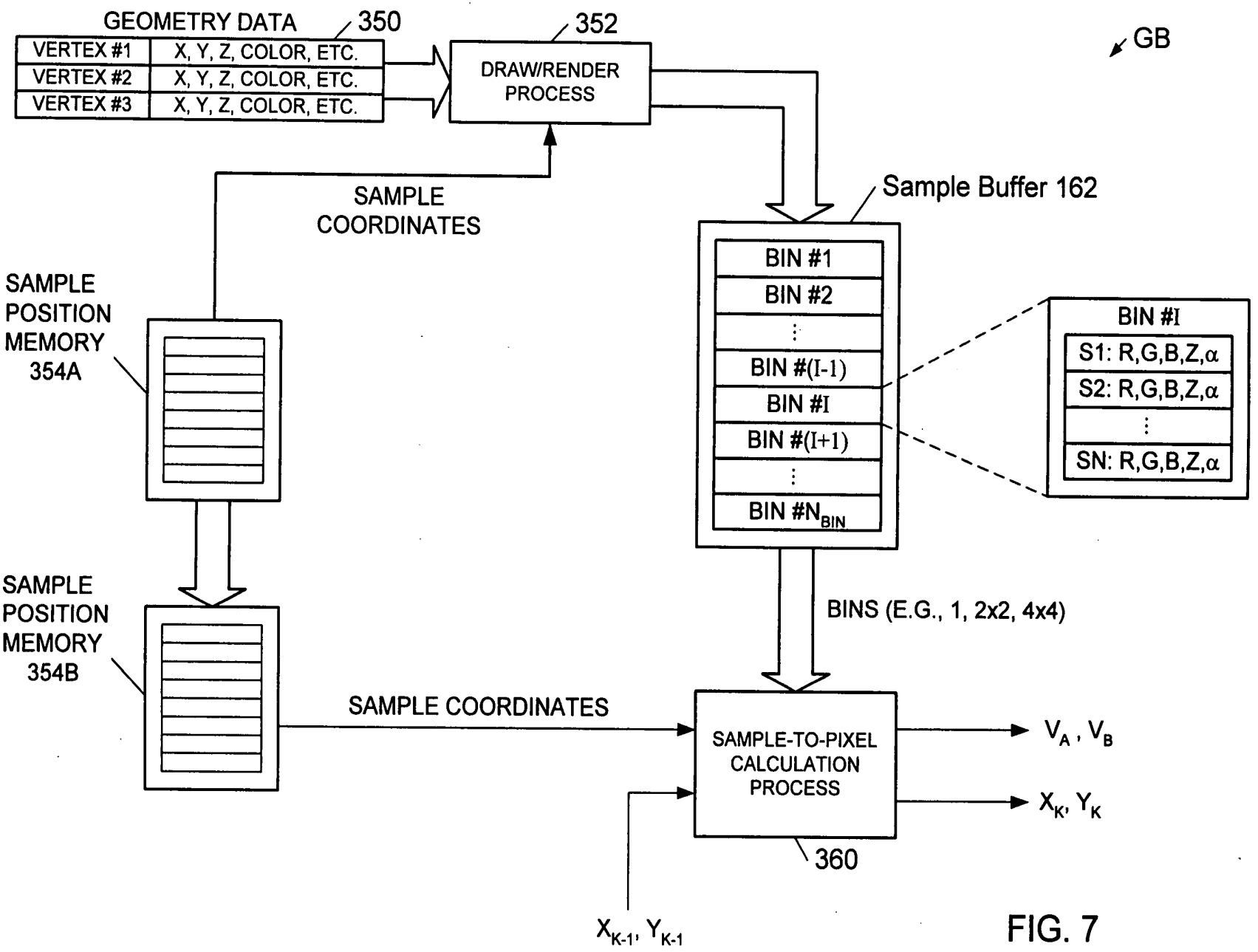
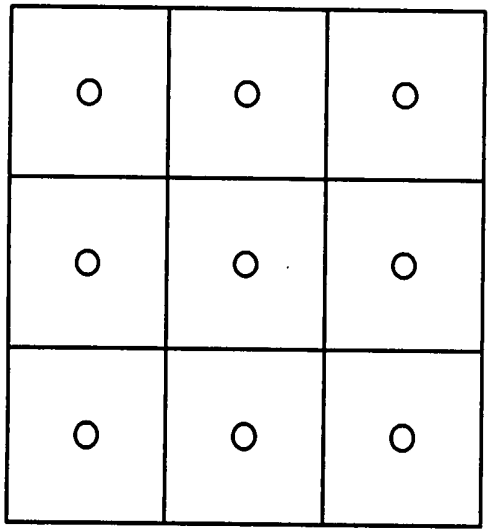
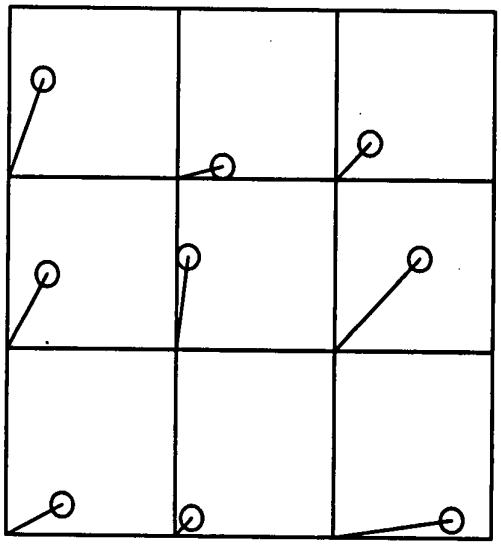

GEOMETRY DATA
350
VERTEX #1
X, Y, Z, COLOR, ETC.
VERTEX #2
X, Y, Z, COLOR, ETC.
VERTEX #3
X, Y, Z, COLOR, ETC.
352
DRAW/RENDER PROCESS
GB
SAMPLE COORDINATES
Sample Buffer 162
BIN #1
BIN #2
BIN #(I-1)
BIN #I
BIN #(I+1)
BIN #N_BIN
BIN #I
S1: R,G,B,Z,α
S2: R,G,B,Z,α
SN: R,G,B,Z,α
SAMPLE POSITION MEMORY 354A
SAMPLE POSITION MEMORY 354B
BINS (E.G., 1, 2x2, 4x4)
SAMPLE COORDINATES
SAMPLE-TO-PIXEL CALCULATION PROCESS
360
V_A , V_B
X_K, Y_K
X_K-1, Y_K-1

FIG. 7



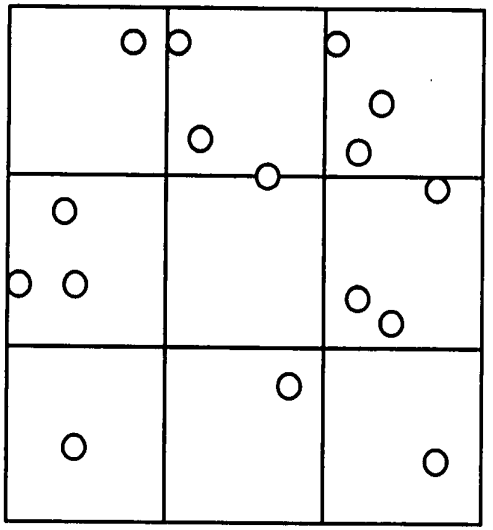
Regular Sample Positioning

190



192

Perturbed Regular Sample Positioning



194

Stochastic Sample Positioning

FIG. 8

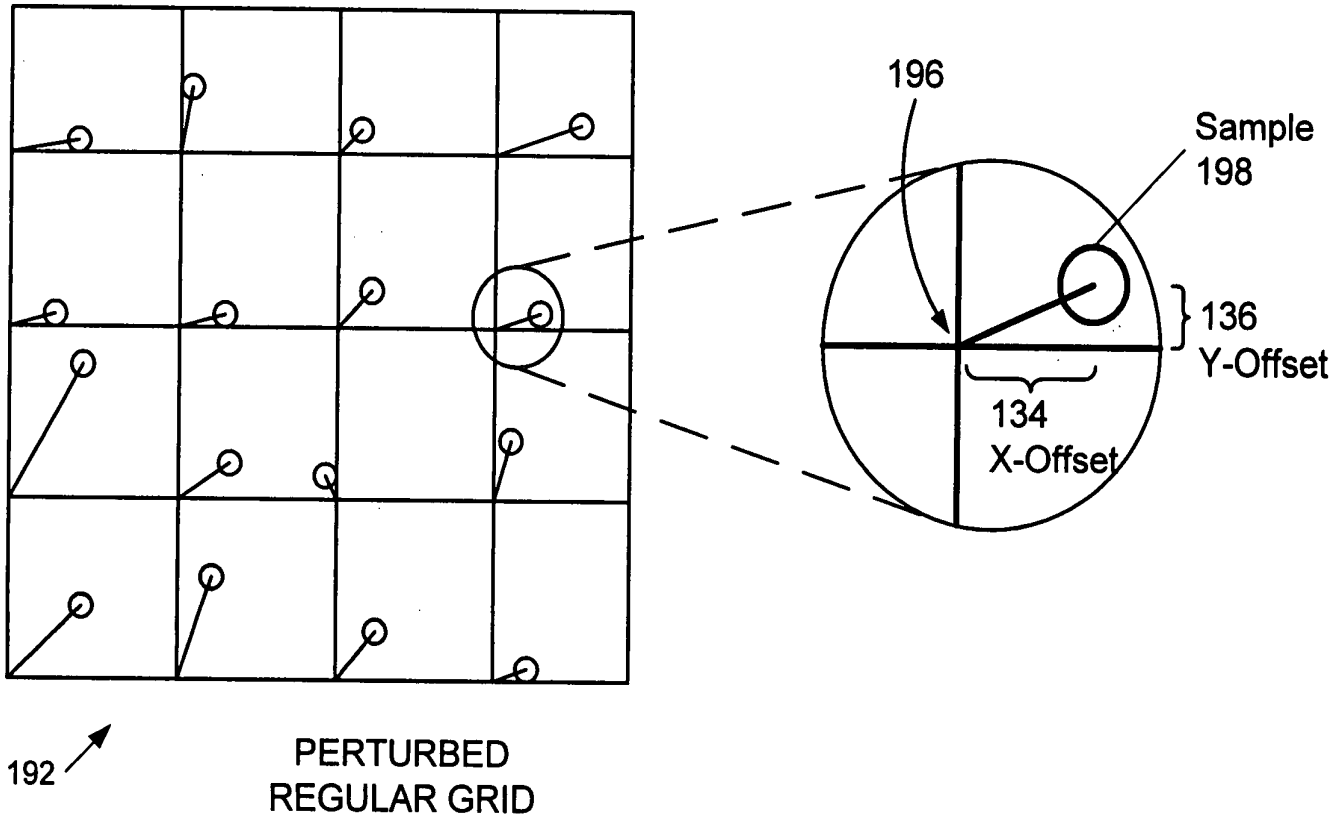

196
Sample
198
136
Y-Offset
134
X-Offset
192
PERTURBED
REGULAR GRID

FIG. 9

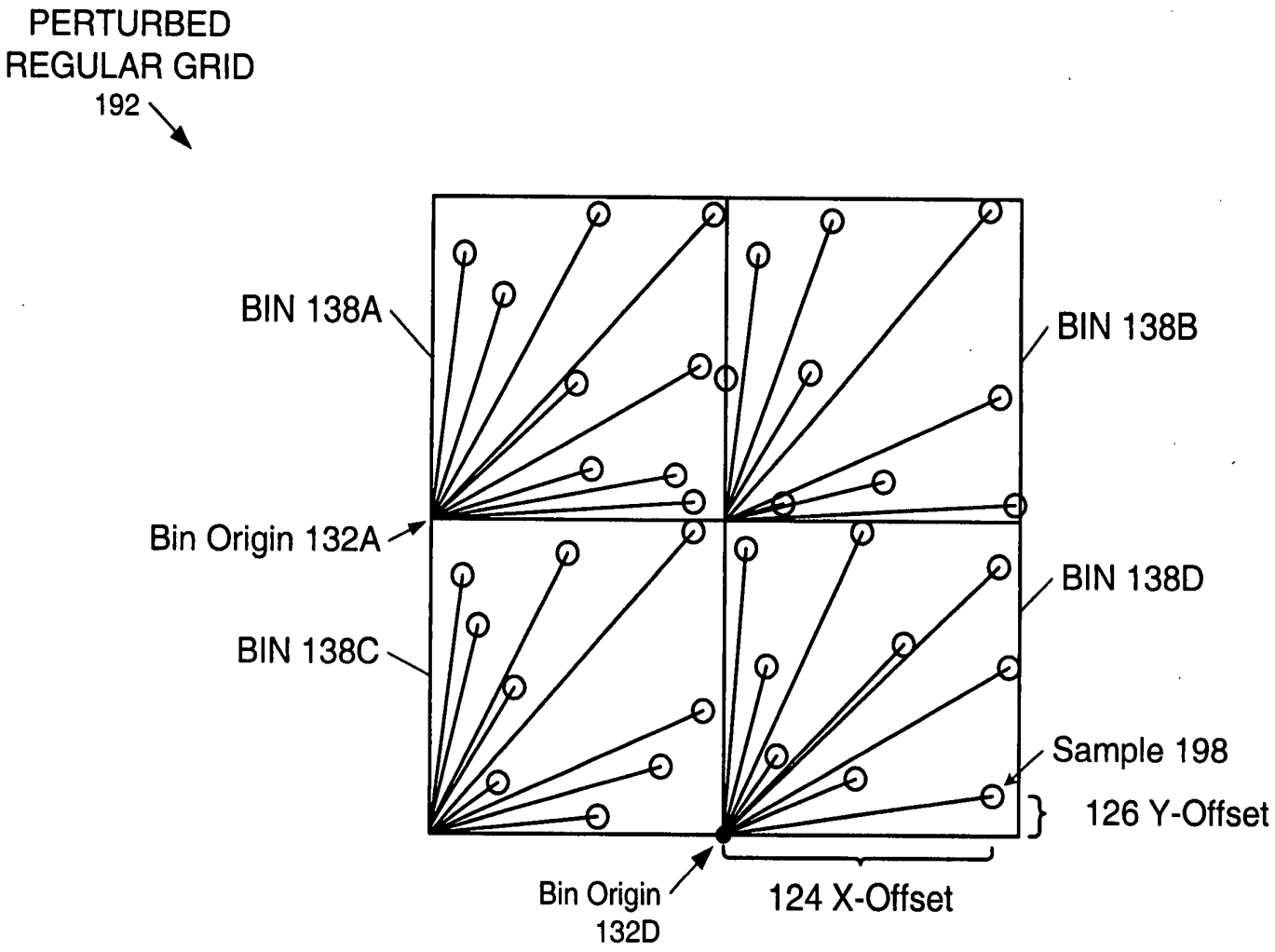

PERTURBED
REGULAR GRID
192
BIN 138A
BIN 138B
Bin Origin 132A
BIN 138C
BIN 138D
Sample 198
126 Y-Offset
Bin Origin
132D
124 X-Offset

FIG. 10

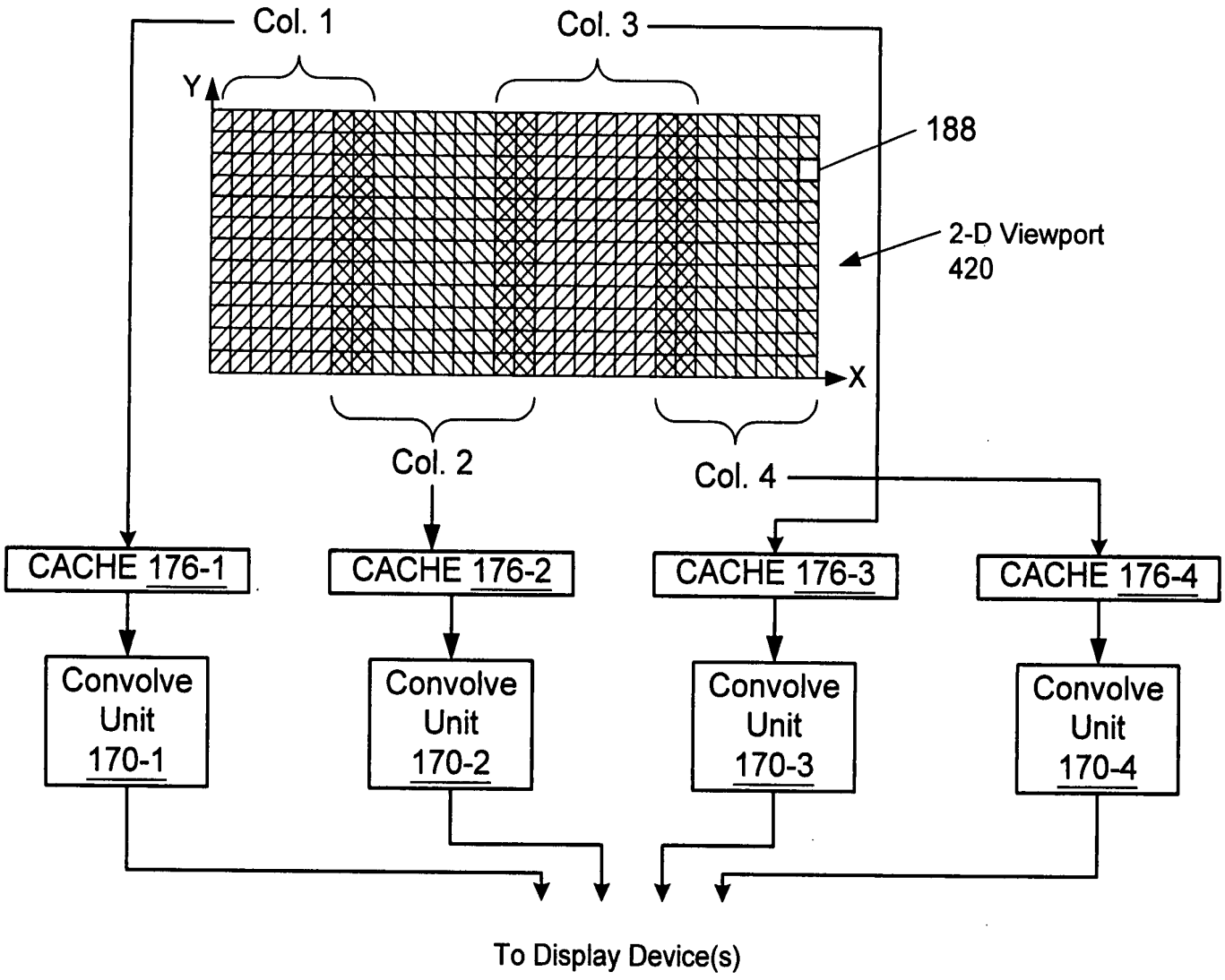

Col. 1
Col. 3
Y
188
2-D Viewport
420
X
Col. 2
Col. 4
CACHE 176-1
CACHE 176-2
CACHE 176-3
CACHE 176-4
Convolve Unit 170-1
Convolve Unit 170-2
Convolve Unit 170-3
Convolve Unit 170-4
To Display Device(s)

FIG. 11

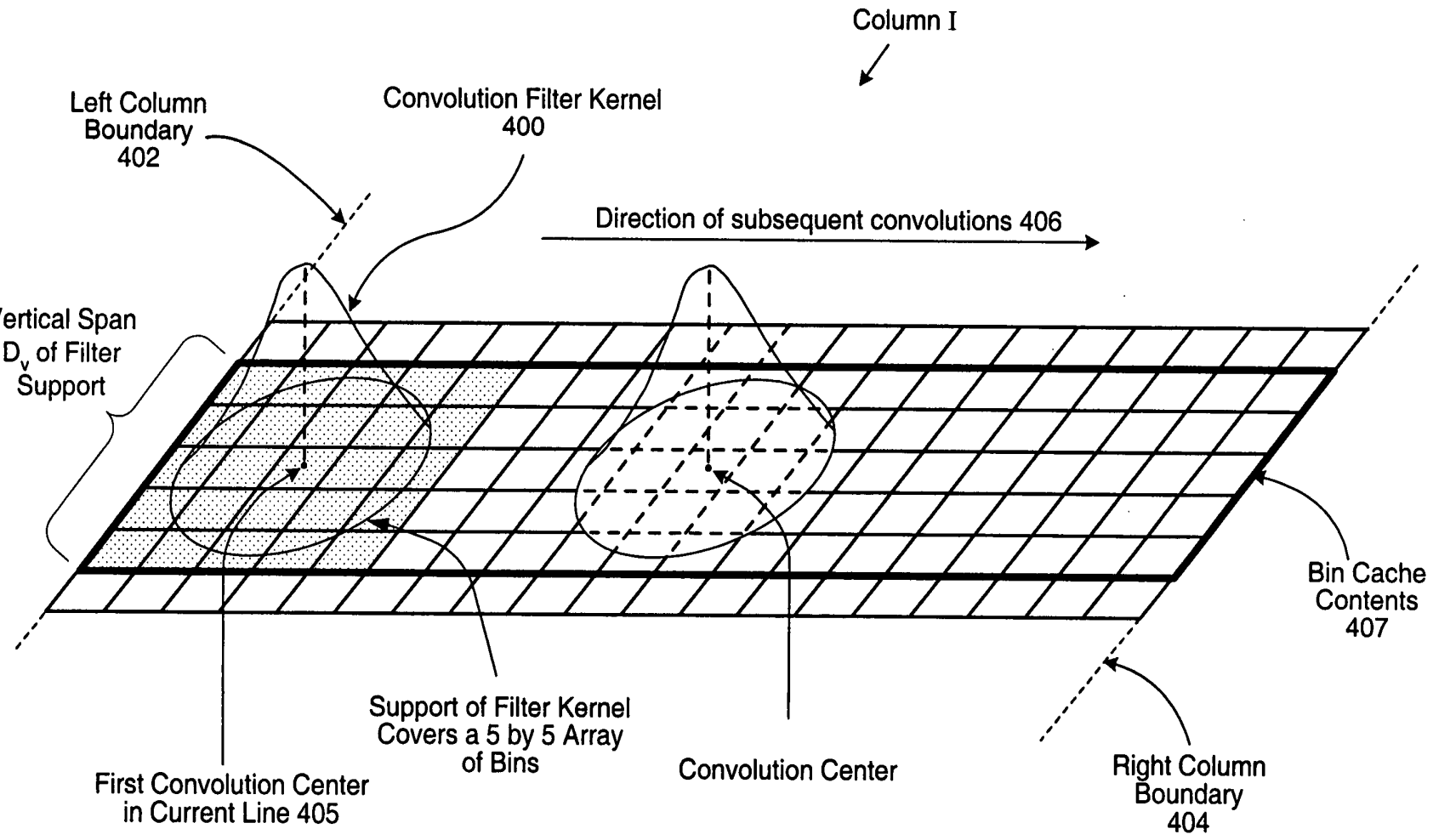

Column I
Left Column Boundary 402
Convolution Filter Kernel 400
Direction of subsequent convolutions 406
Vertical Span D_v of Filter Support
Bin Cache Contents 407
Support of Filter Kernel Covers a 5 by 5 Array of Bins
First Convolution Center in Current Line 405
Convolution Center
Right Column Boundary 404

FIG. 12A

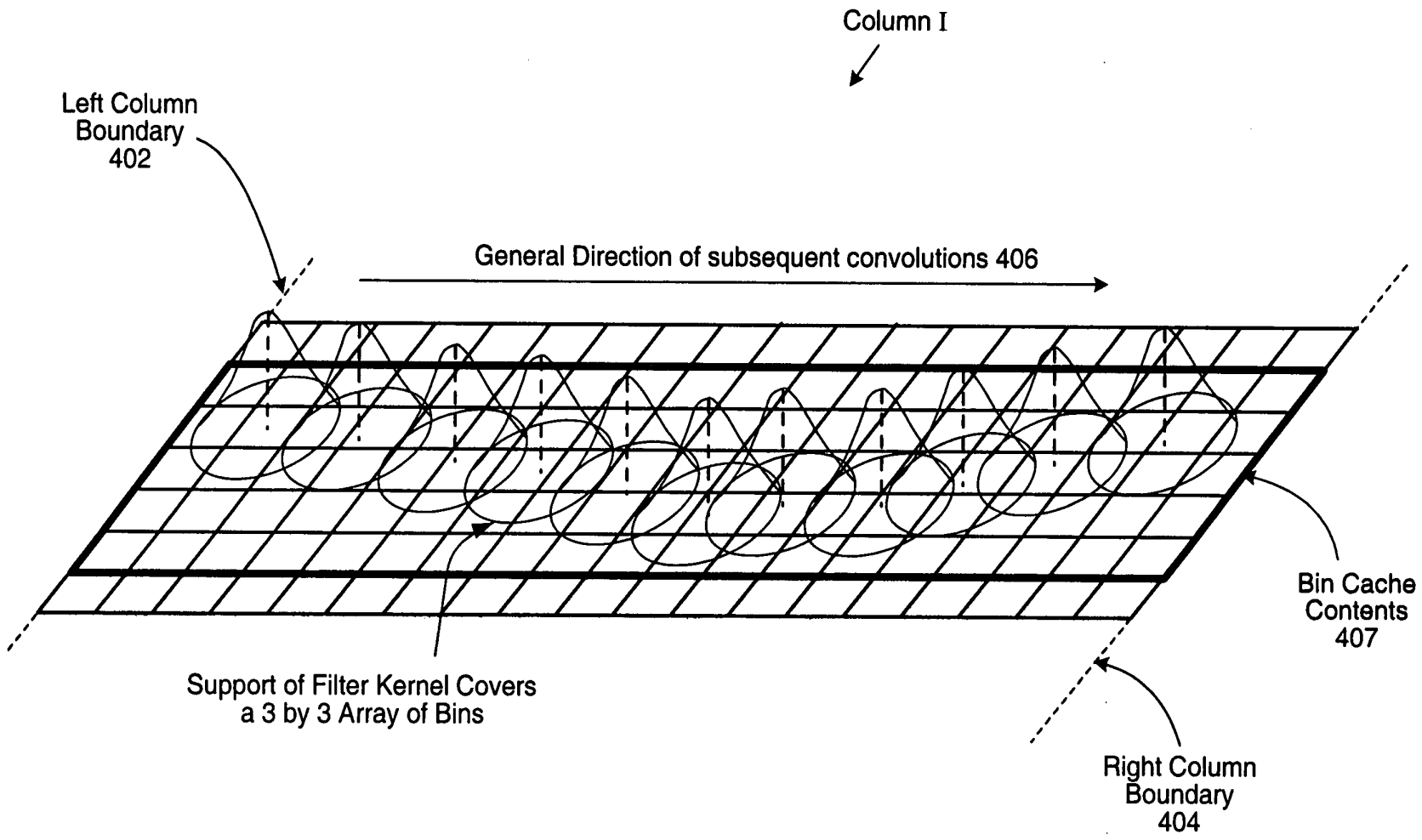

Column I
Left Column Boundary 402
General Direction of subsequent convolutions 406
Bin Cache Contents 407
Support of Filter Kernel Covers a 3 by 3 Array of Bins
Right Column Boundary 404

FIG. 12B

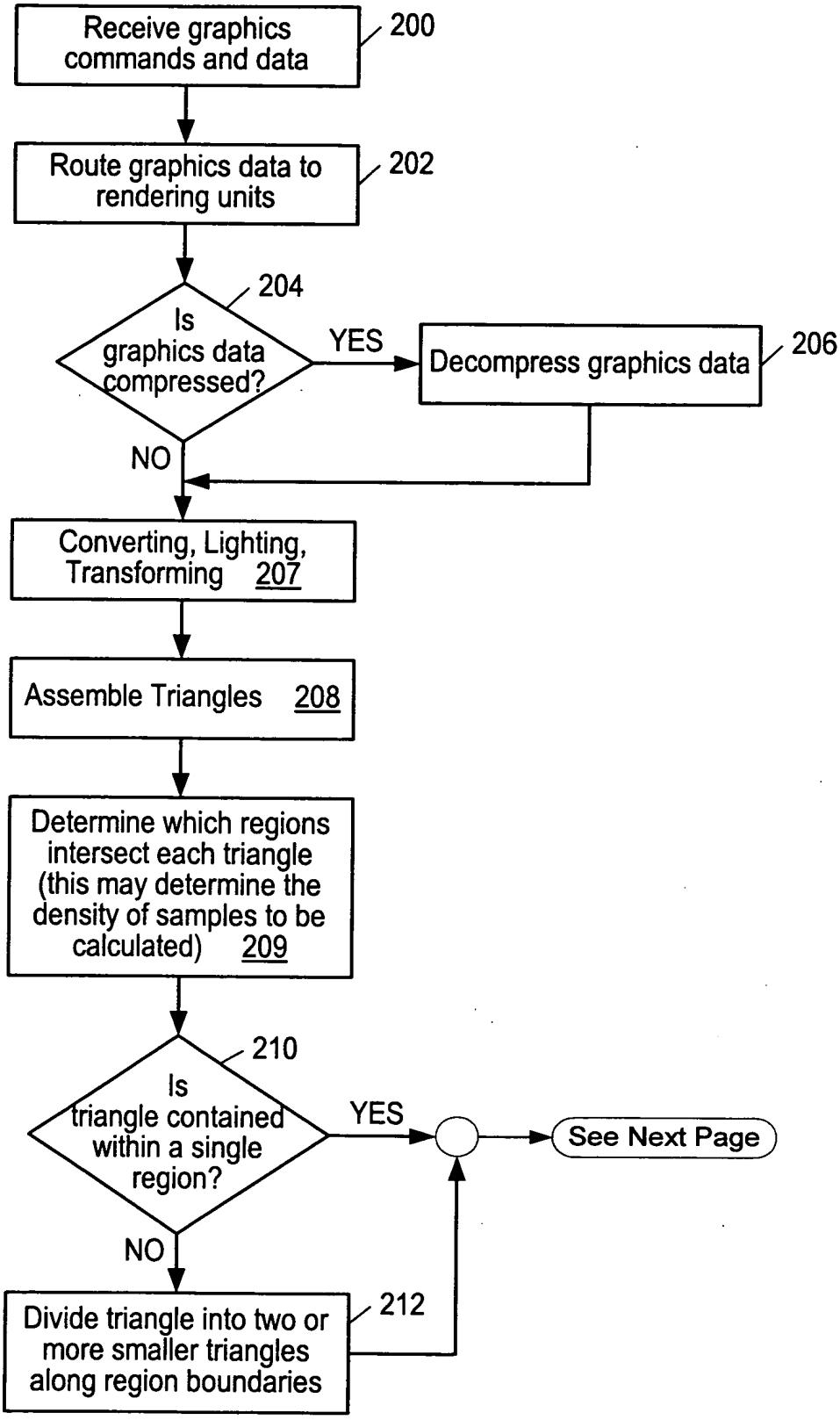

Receive graphics commands and data
200
Route graphics data to rendering units
202
204
Is graphics data compressed?
YES
Decompress graphics data
206
NO
Converting, Lighting, Transforming 207
Assemble Triangles 208
Determine which regions intersect each triangle (this may determine the density of samples to be calculated) 209
210
Is triangle contained within a single region?
YES
See Next Page
NO
Divide triangle into two or more smaller triangles along region boundaries
212

FIG. 13A

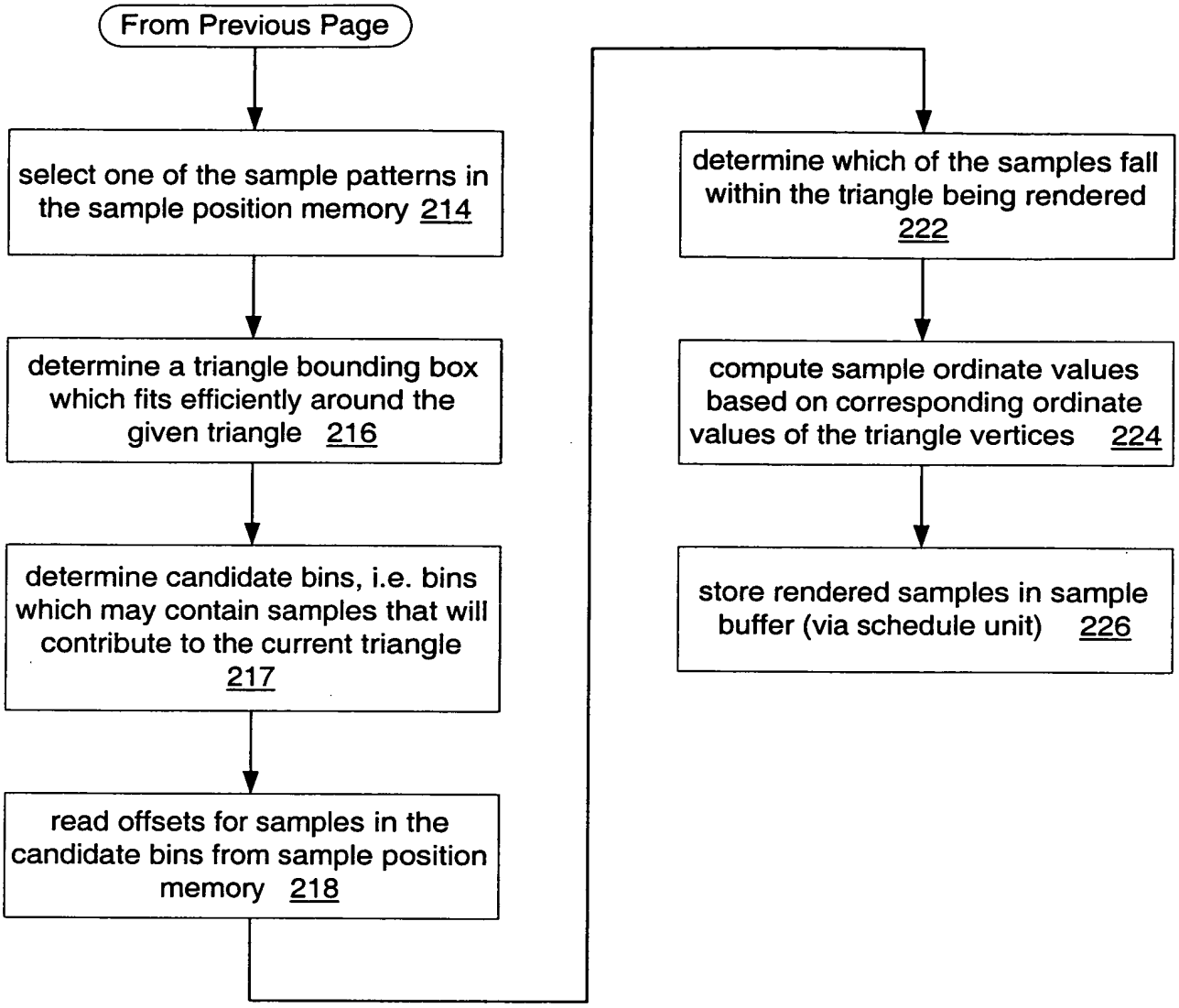

From Previous Page
select one of the sample patterns in the sample position memory 214
determine a triangle bounding box which fits efficiently around the given triangle 216
determine candidate bins, i.e. bins which may contain samples that will contribute to the current triangle 217
read offsets for samples in the candidate bins from sample position memory 218
determine which of the samples fall within the triangle being rendered 222
compute sample ordinate values based on corresponding ordinate values of the triangle vertices 224
store rendered samples in sample buffer (via schedule unit) 226

FIG. 13B

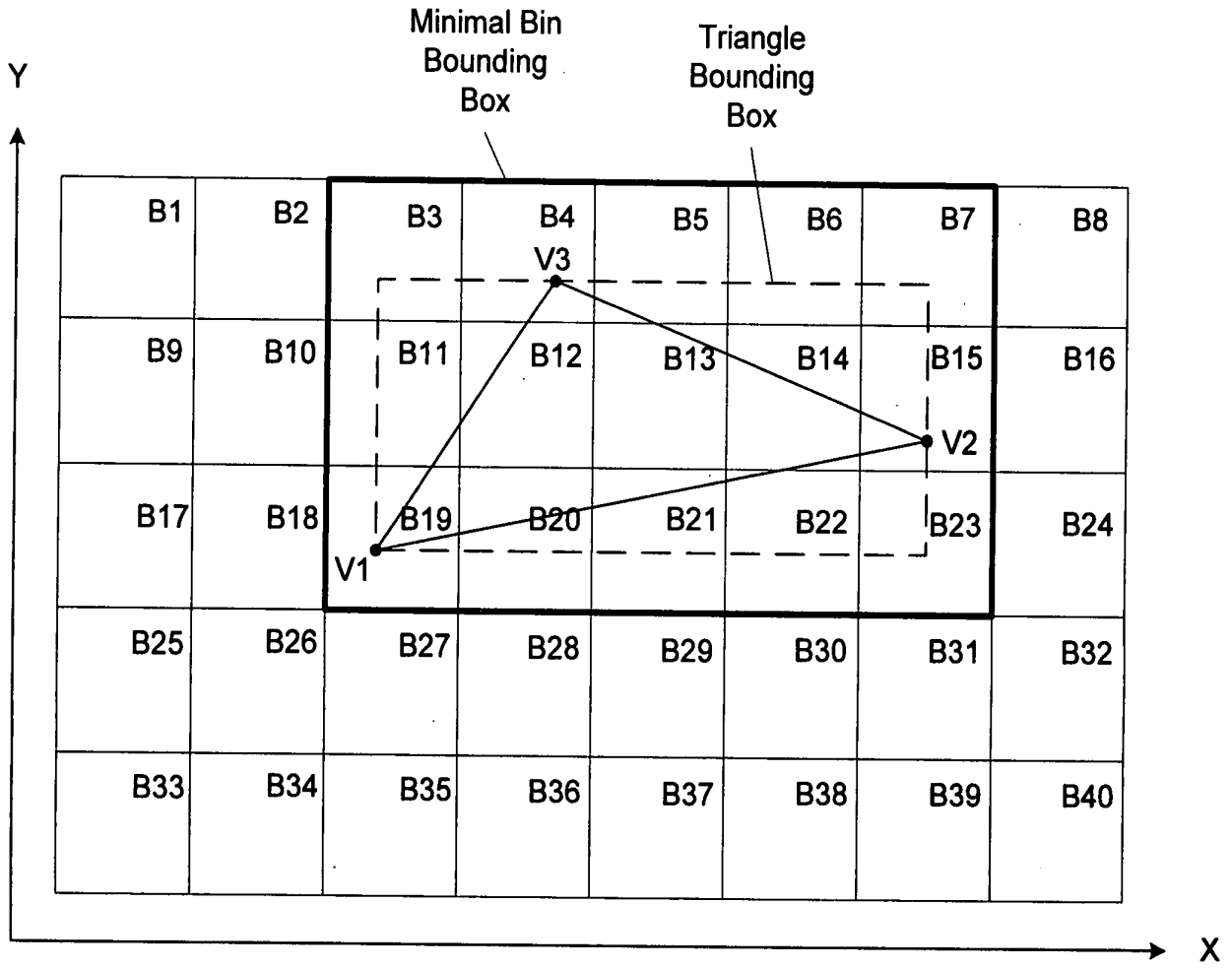

Minimal Bin Bounding Box
Triangle Bounding Box
Y
X
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
B26
B27
B28
B29
B30
B31
B32
B33
B34
B35
B36
B37
B38
B39
B40
V1
V2
V3

Fig. 13C

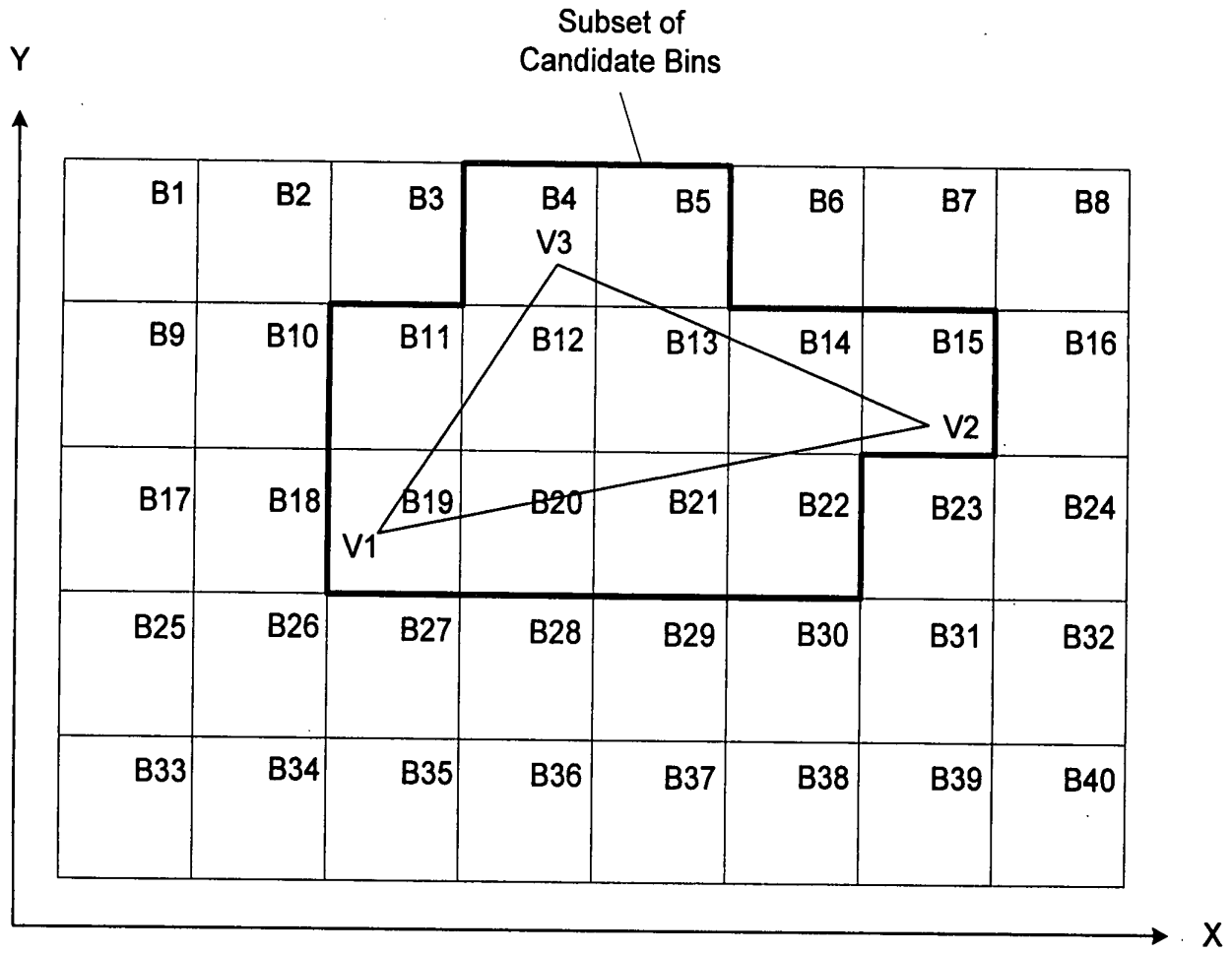
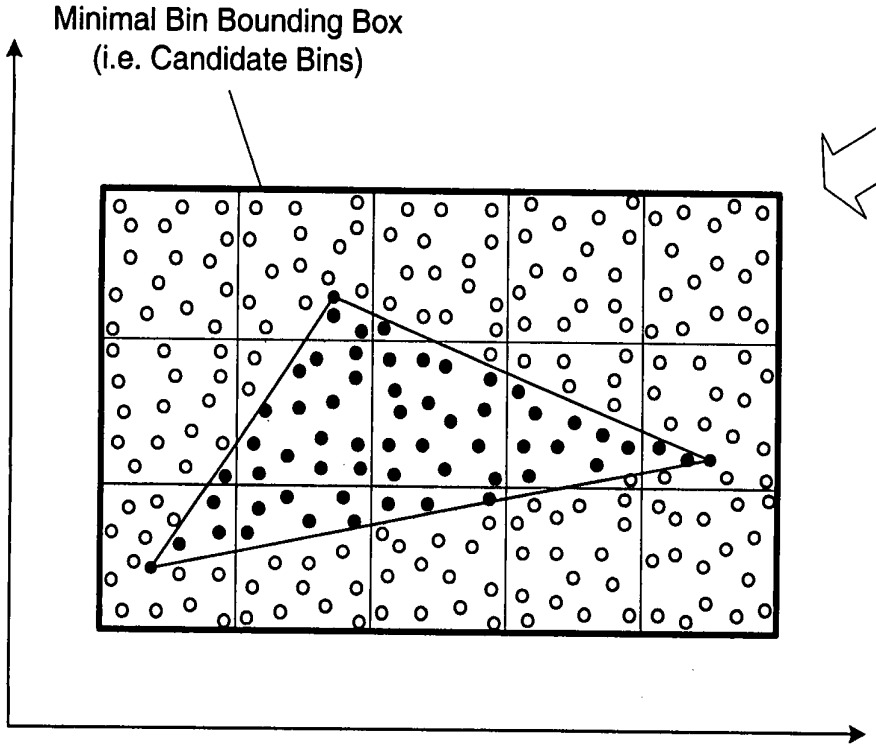

Subset of
Candidate Bins
Y
X
B1
B2
B3
B4
B5
B6
B7
B8
V3
B9
B10
B11
B12
B13
B14
B15
B16
V2
B17
B18
B19
B20
B21
B22
B23
B24
V1
B25
B26
B27
B28
B29
B30
B31
B32
B33
B34
B35
B36
B37
B38
B39
B40

Fig. 13D



Bin Identifications

B3	B4	B5	B6	B7
B11	B12	B13	B14	B15
B19	B20	B21	B22	B22

Fig. 13E

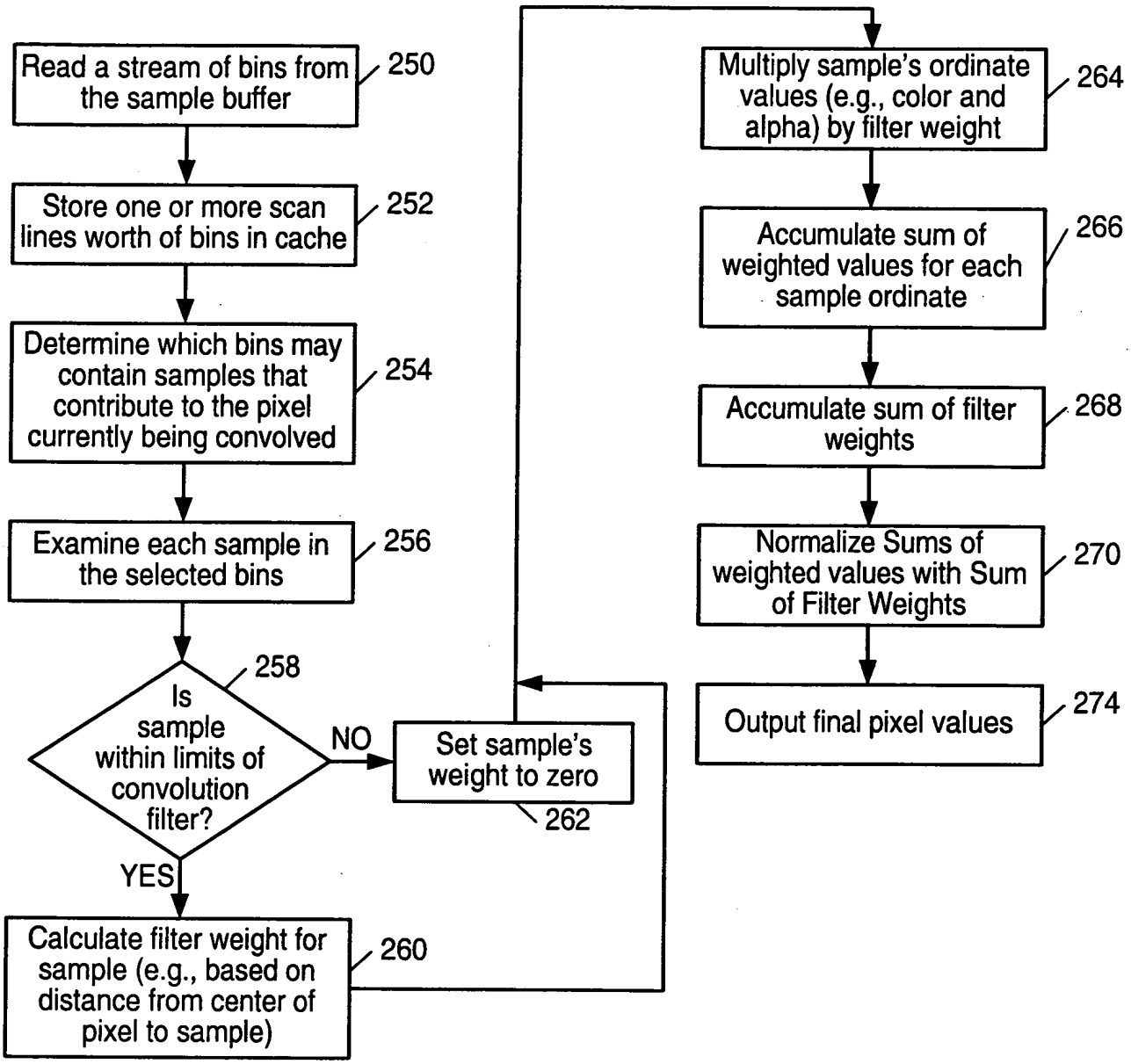
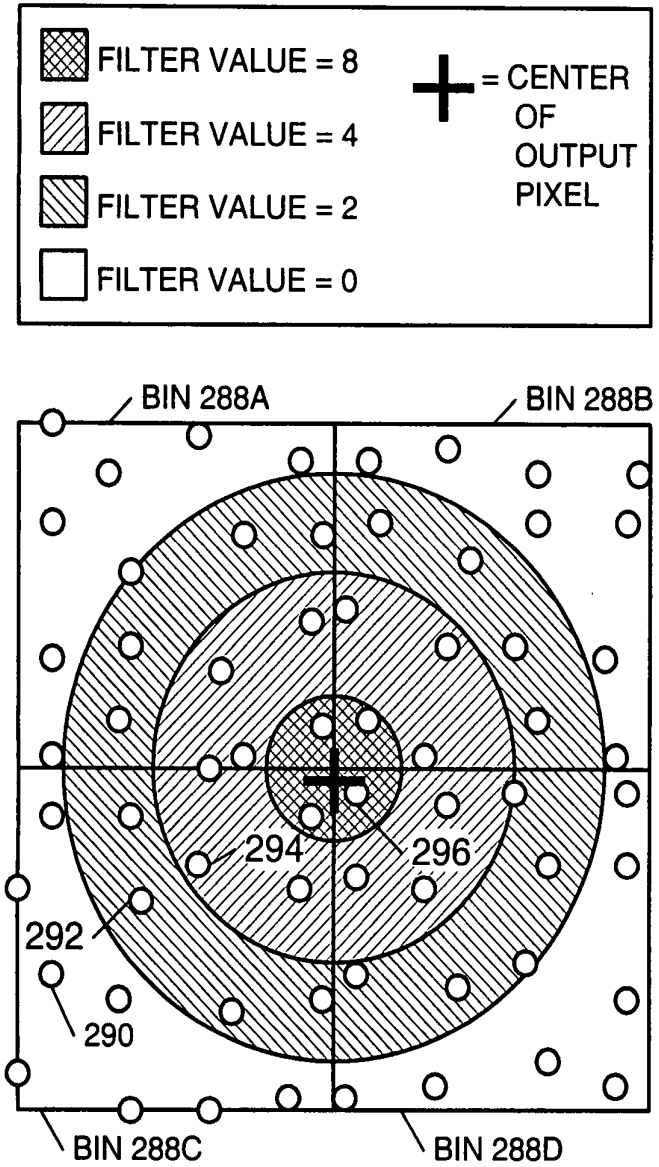

Read a stream of bins from the sample buffer
250
Store one or more scan lines worth of bins in cache
252
Determine which bins may contain samples that contribute to the pixel currently being convolved
254
Examine each sample in the selected bins
256
Is sample within limits of convolution filter?
258
NO
Set sample's weight to zero
262
YES
Calculate filter weight for sample (e.g., based on distance from center of pixel to sample)
260
Multiply sample's ordinate values (e.g., color and alpha) by filter weight
264
Accumulate sum of weighted values for each sample ordinate
266
Accumulate sum of filter weights
268
Normalize Sums of weighted values with Sum of Filter Weights
270
Output final pixel values
274

FIG. 14

06627 "/6225760



300

Sample 290

R = 120 FILTER
G = 200 VALUE = 0
B = 40
A = 150

302

Sample 292

R = 140 FILTER
G = 180 VALUE = 2
B = 50
A = 160

304

Sample 294

R = 150 FILTER
G = 170 VALUE = 4
B = 50
A = 180

306

Sample 296

R = 140 FILTER
G = 170 VALUE = 8
B = 60
A = 190

308

NORMALIZATION
VALUE = 0+2+4+8 = 14

310

UNNORMALIZED PIXEL VALUES

R = 120*0
+140*2
+150*4
+140*8 = 2000

G = 200*0
+180*2
+170*4
+170*8 = 2400

B = 40*0
+50*2
+50*4
+60*8 = 780

A = 150*0
+160*2
+180*4
+190*8 = 2560

312

NORMALIZED PIXEL VALUES

R = 2000 / 14 = 142.9
G = 2400 / 14 = 171.4
B = 780 / 14 = 55.7
A = 2560 / 14 = 175.7

FIG. 15

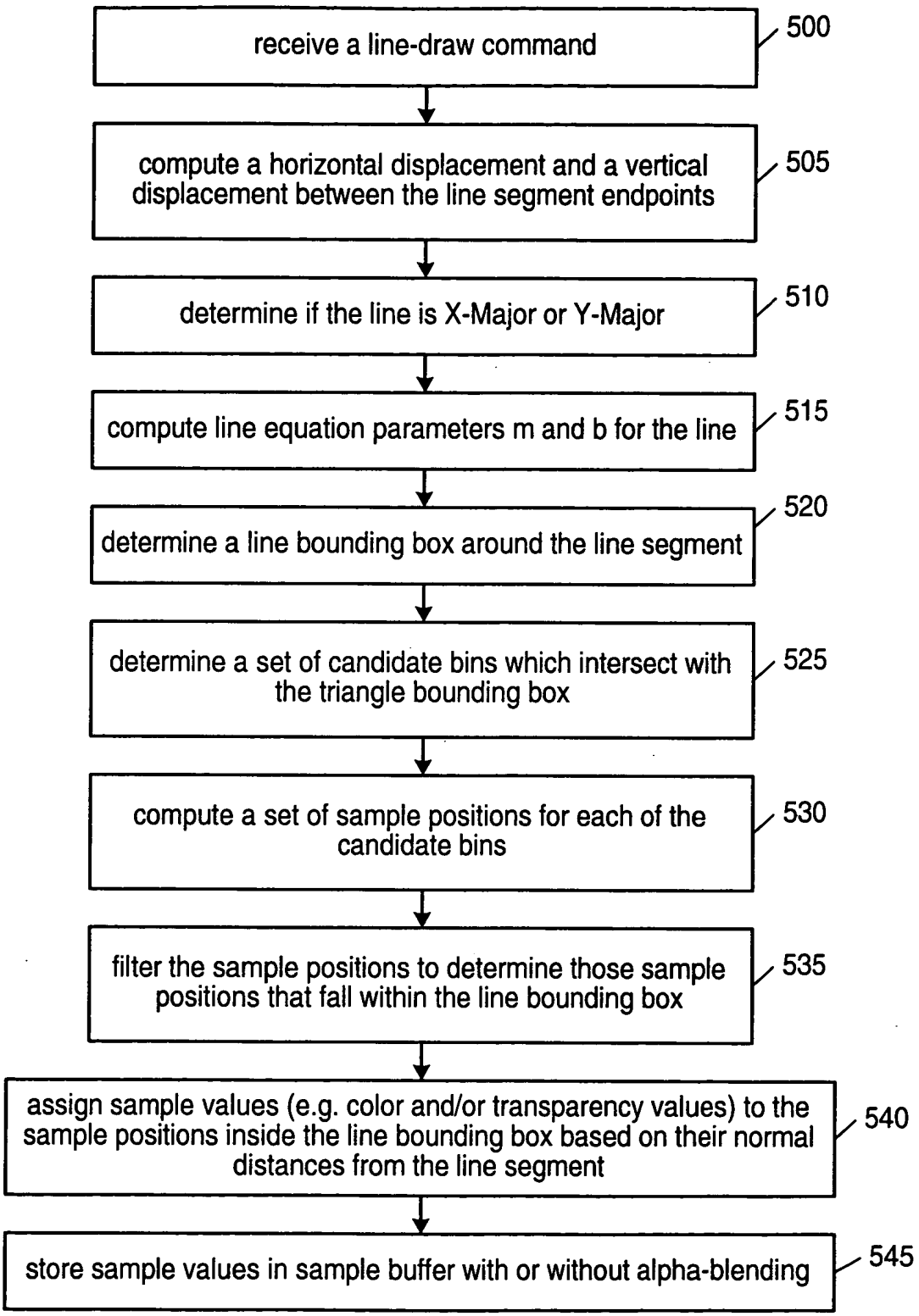

receive a line-draw command
500
compute a horizontal displacement and a vertical displacement between the line segment endpoints
505
determine if the line is X-Major or Y-Major
510
compute line equation parameters m and b for the line
515
determine a line bounding box around the line segment
520
determine a set of candidate bins which intersect with the triangle bounding box
525
compute a set of sample positions for each of the candidate bins
530
filter the sample positions to determine those sample positions that fall within the line bounding box
535
assign sample values (e.g. color and/or transparency values) to the sample positions inside the line bounding box based on their normal distances from the line segment
540
store sample values in sample buffer with or without alpha-blending
545

FIG. 16

06227 26225260

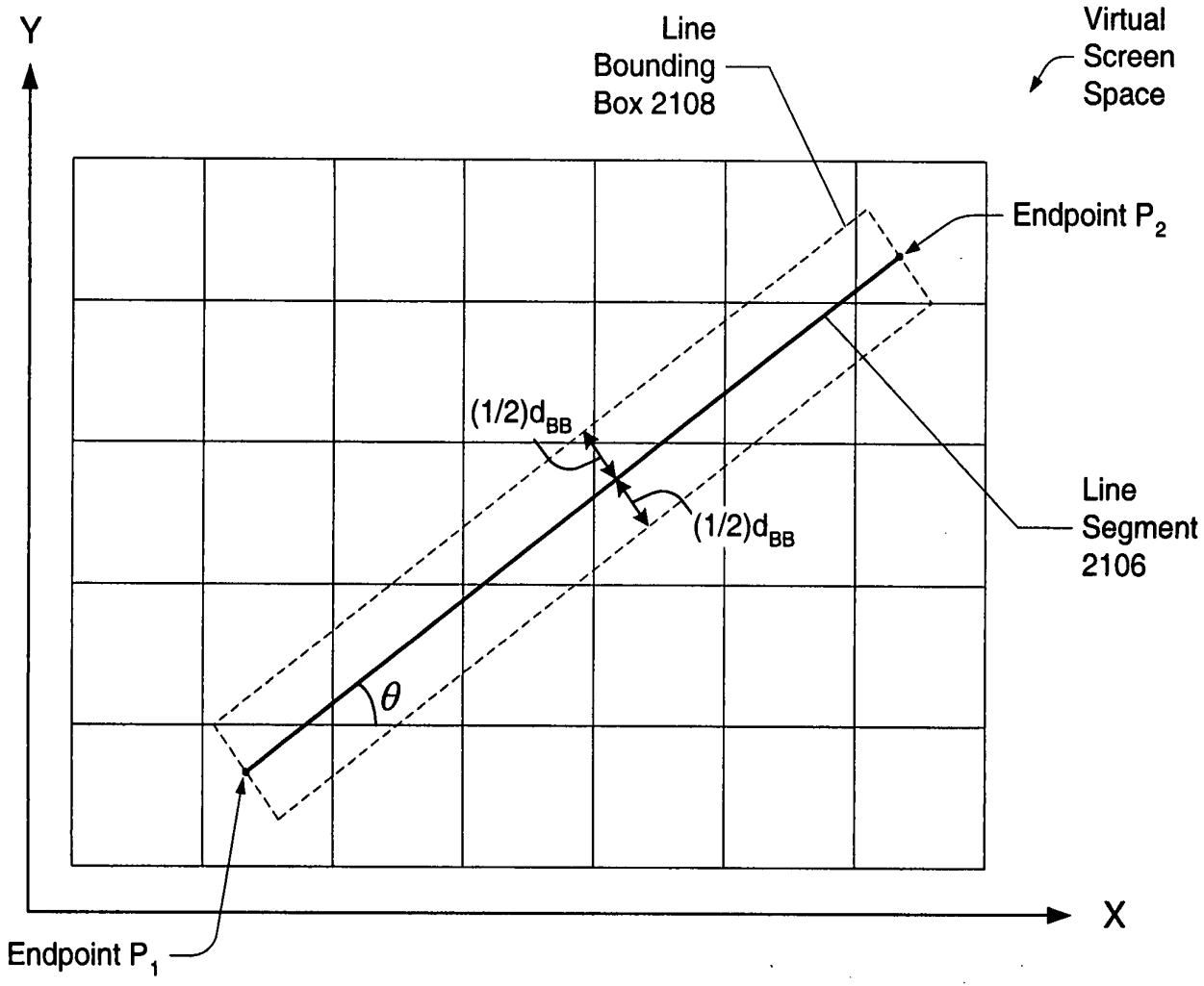


Fig. 17A

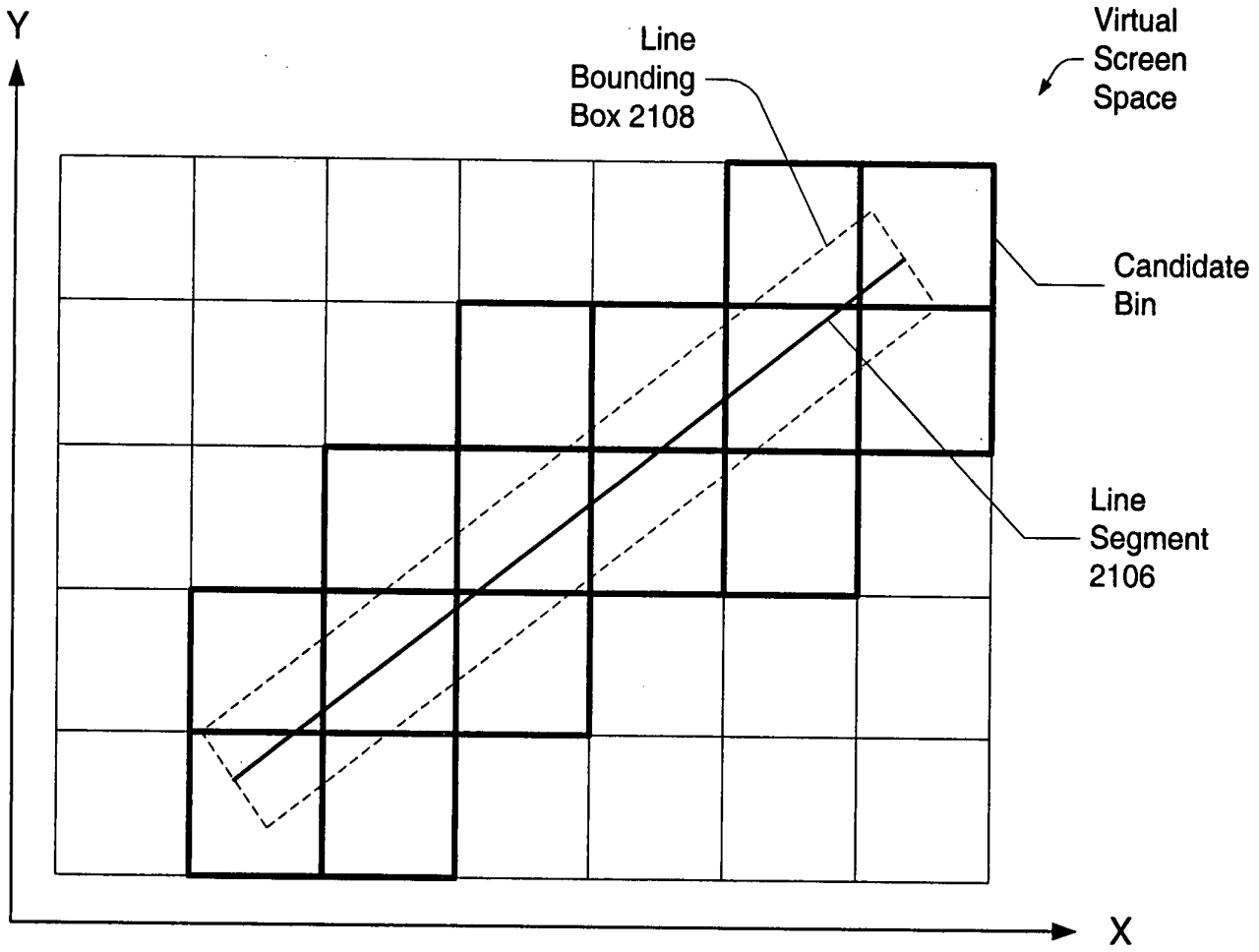

Y
Line
Bounding
Box 2108
Virtual
Screen
Space
Candidate
Bin
Line
Segment
2106
X

Fig. 17B

006227 26225260

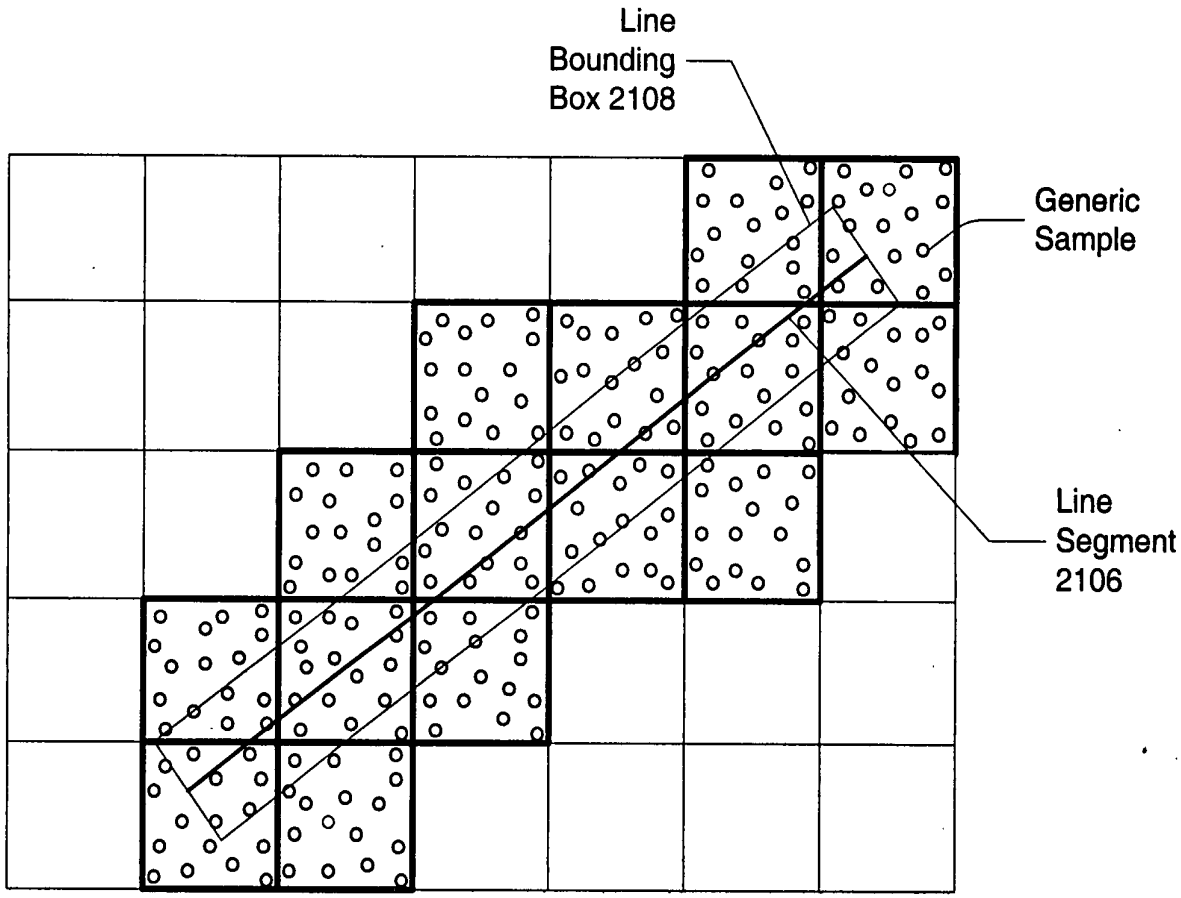


Fig. 17C

006327" 26225760

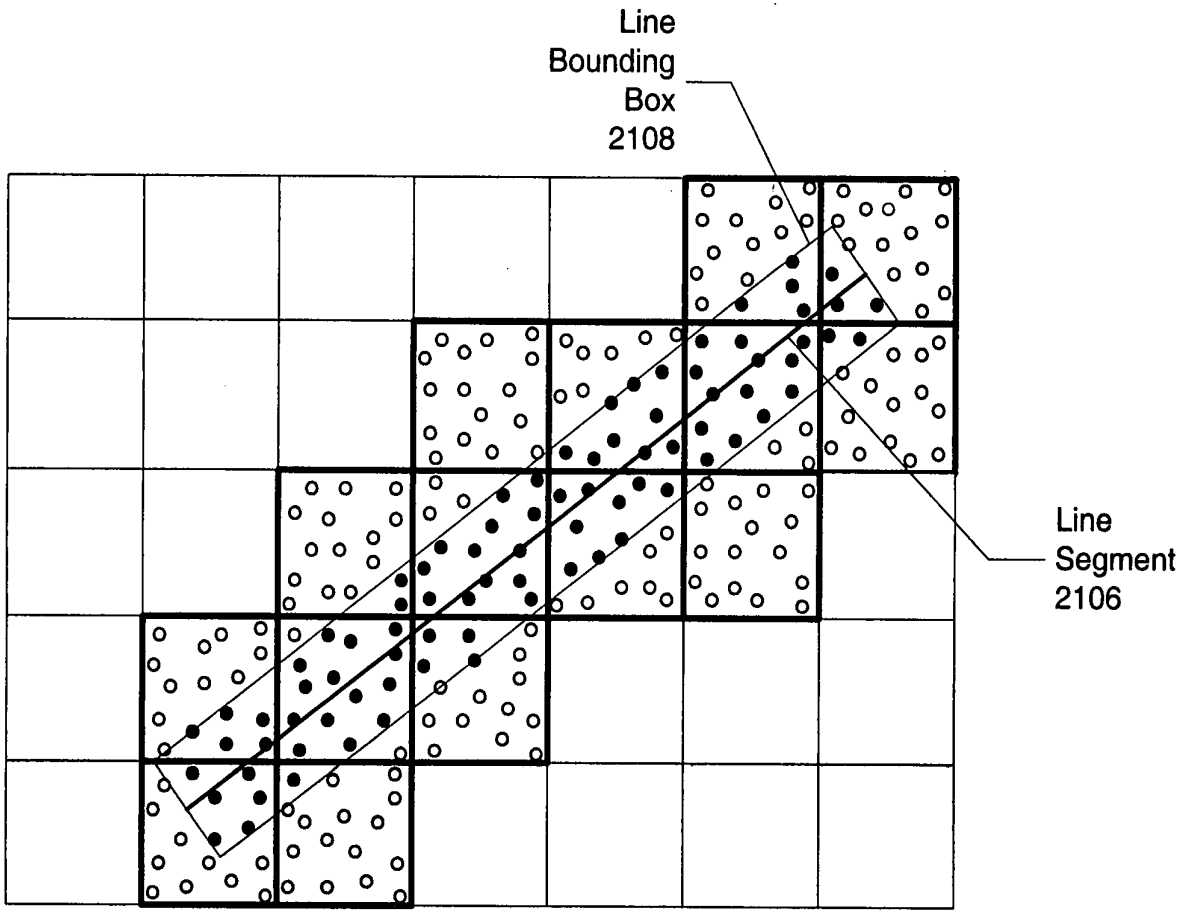


Fig. 17D

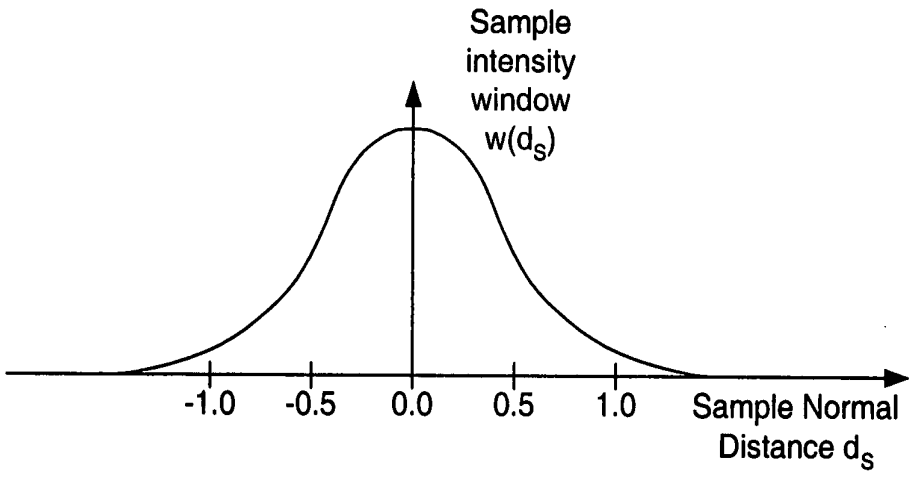

Sample
intensity
window
w(d_s)
-1.0
-0.5
0.0
0.5
1.0
Sample Normal
Distance d_s

Fig. 17E

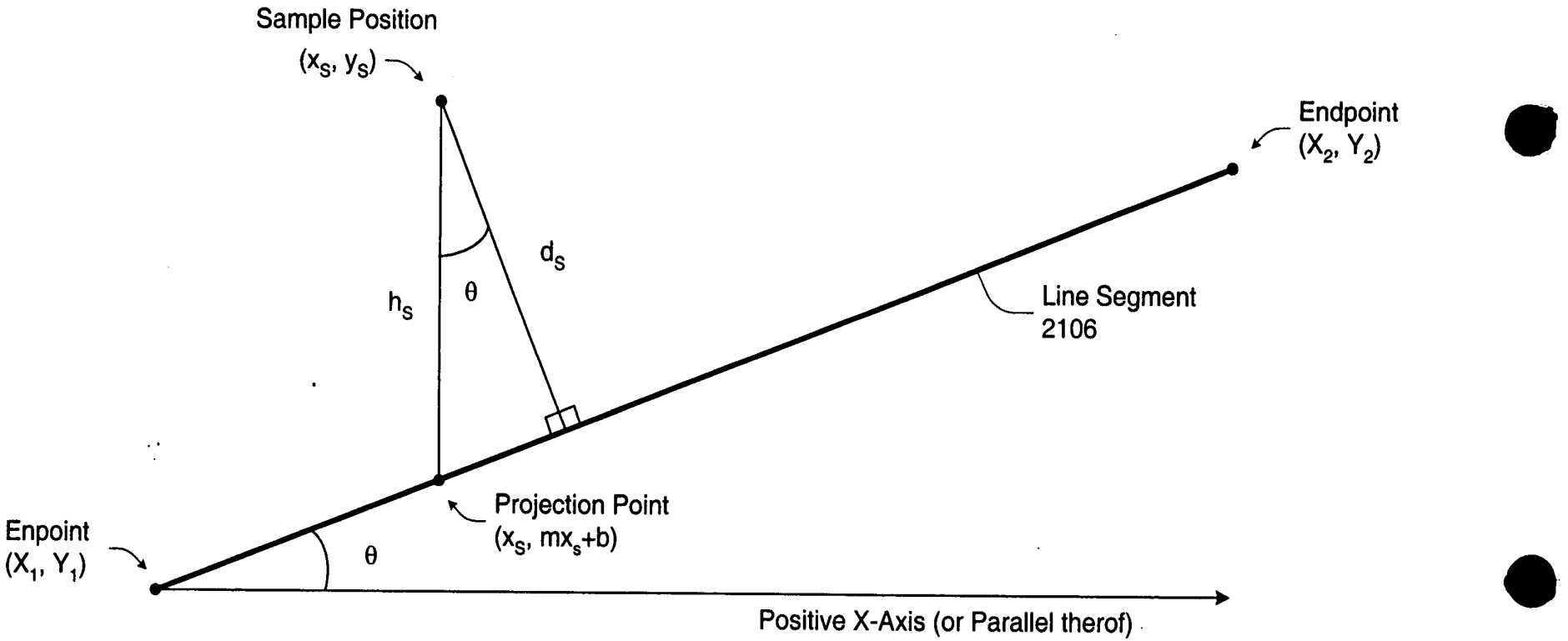

Sample Position
(x_S, y_S)
Endpoint
(X_2, Y_2)
d_S
h_S
θ
Line Segment
2106
Projection Point
(x_S, mx_s+b)
Enpoint
(X_1, Y_1)
θ
Positive X-Axis (or Parallel therof)

Fig. 17F

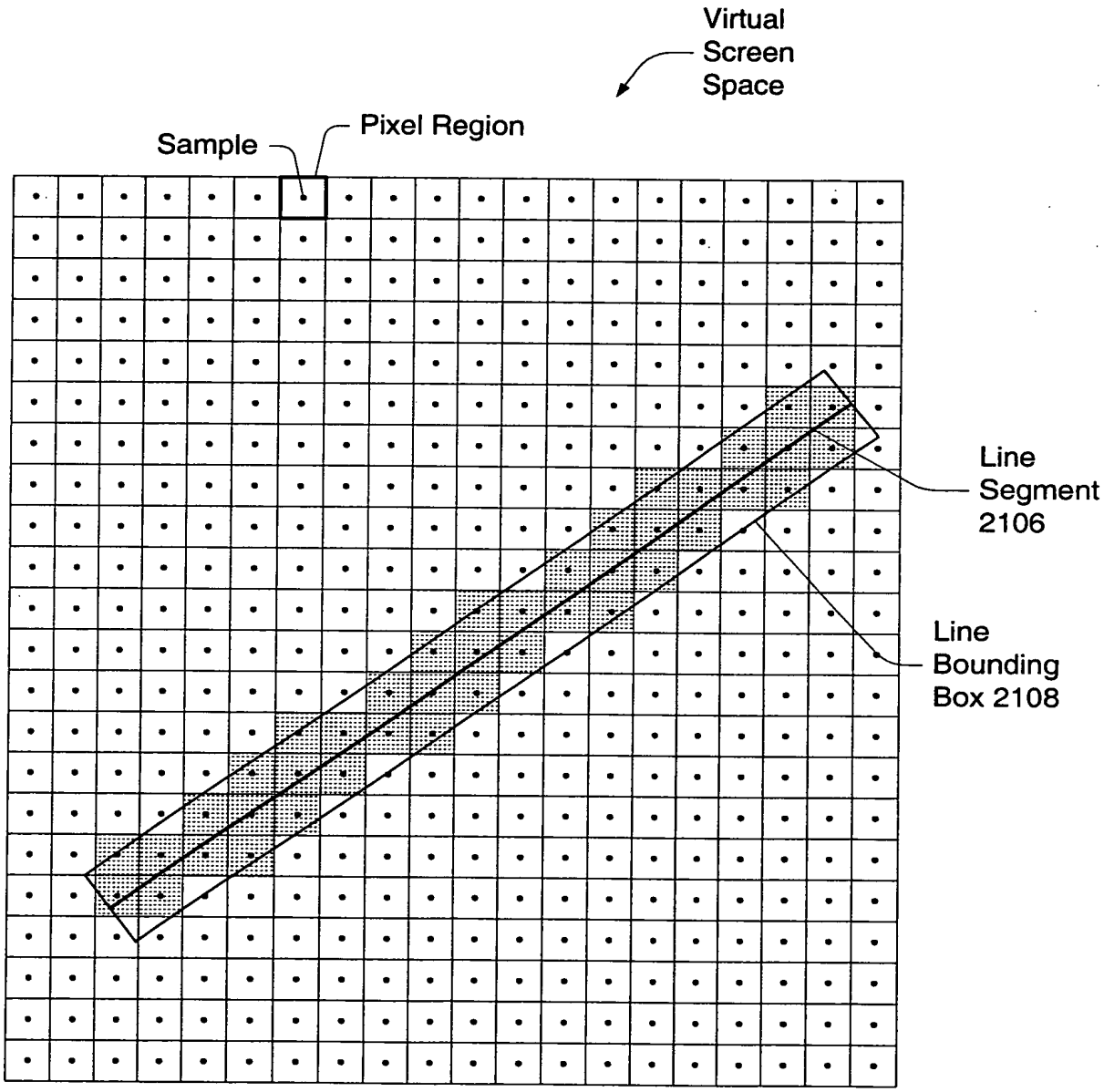

Virtual
Screen
Space
Sample
Pixel Region
Line
Segment
2106
Line
Bounding
Box 2108

Fig. 18